Eigenvoice Adaptation of Acoustic sub-word models

G. Sriram and T. V. Sreenivas*
Department of Electrical Communication Engineering
Indian Institute of Science, Bangalore 560 012, India.
Phone: +91 80 2360 2167, Fax: +91 80 2360 0683.
e-mail:ram@ece.iisc.ernet.in, *tvsree@ece.iisc.ernet.in

EDICS: SPE-RECO, SPE-ADAP

Abstract

The use of acoustic sub-words for speech recognition provides interesting challenges like clustering of speech into sub-words, modeling of these units, lexicon building etc. The earlier techniques proposed for segmentation and labeling of speech into acoustic sub-word units (ASWUs), do not capture the dynamics of the segments properly. Hence, speaker independent ASR systems using acoustic units have been less successful. In this paper, we use an iterative HMM based clustering technique for training the ASWU models. A first order Markov lexicon or a speaker dependent lexicon is used for the recognition tasks. Further, adaptation of the ASWU models for new speakers has been done using the eigenvoice approach. From the speakers in the training corpus, a set of eigenvoices are obtained, which contain the sub-word HMM parameters to be adapted. For adaptation with both labeled as well as un-labeled adaptation data, two adaptation procedures, based on sub-word eigenvoices, are proposed. The issues of lexical dependence/independence for adaptation is addressed in these adaptation schemes. In the algorithms for building the SI ASWU models as well as that for adapting these models to a new speaker, the freedom of soft labels to ASWUs is efficiently utilized. Finally, we report the results for experiments performed on a small vocabulary IWR task using BPL database.

Index Terms

Acoustic sub-words, speaker adaptation, sub-word eigenvoices.

I. INTRODUCTION

Popular ASR systems use sub-words units (SWUs) based upon linguistic description of the language. Typical examples of linguistic sub-word units (LSWUs) are phonemes, diphones, triphones and syllables. The LSWUs (such as phonemes) have the advantage that the word lexicon is available in a ready-made form from a standard dictionary and hence, widely used in ASR systems. There is, however, a major problem when it comes to correctly detecting and identifying these units. The success of the LSWU based systems is entirely dependent on the accuracy of the manual segmentation of the training data. If only a small amount of manually segmented data is available, some form of bootstrapping procedure is then used to automatically segment a larger set of training data in order to have sufficient data for training the sub-word models. Since linguistic boundaries are often ill-defined in the acoustic signal, the segmentation and the resulting models are not guaranteed to be consistent with their original linguistic definitions.

Acoustic sub-words for ASR has been interesting research topic for nearly two decades now. As the use of ASWUs avoids the requirement of manual speech segmentation and labeling, these systems are attractive. But, the problems of automatic speech clustering, ASWU modeling and lexicon building need to addressed in an efficient way to build recognition systems with practically acceptable performances. The earliest approach to ASWU based systems use a spectrally defined variation contour for defining the acoustic units ([1], [2]). The variation contour gives good sub-word segmentation when employed for single speaker tasks. But, since a single quantity is insufficient to represent spectral variation in multi-speaker environments, speaker independent recognition tasks have not been successful using this.

Other approaches to acoustic unit segmentation is using inter frame correlations ([3] - [5]). In this approach, Karhunen-Loeve transformation (KLT) is used to extract and efficiently represent the highly correlated structure of the spectral envelope. This is in contrast to frame based approaches, which ignores these correlations. signal. It also incorporates linguistical knowledge into a mathematical framework to determine time varying acoustic-phonetic features, which means that the units derived are not purely acoustic in nature. [6] uses changes in AR parameters of speech to automatically segment the speech signal into sub-words. But, the method ignores the fact that all speech sounds cannot be modeled by AR parameters. A global acoustic model consisting of intra segment acoustic models and inter-segment

transition rules is used in [7]. The more recent approach ([8]- [10]) employ Maximum Likelihood (ML) segmentation followed by segment centroid based clustering methods. HMMs are used for modeling the segment clusters. The main disadvantage in this technique dynamics of the acoustic segment are not captured by representing it using a centroid.

In this paper, we use an iterative HMM based clustering technique for obtaining the ASWU models. Constrained length ML segmentation ([8]) is used to obtain the acoustic segments and VQ based clustering, without leaving the dynamics within a segment, gives the initial set of clusters. Sub-word HMMs are built and then used to obtain new clusters. This mechanism is iterated to obtain a set of sub-word models. The soft labels to acoustic sub-words is exploited in this iterative HMM training method. Our technique is similar to [11], except that the initial clustering is different and that the base forms are not explicitly used. Lexicon building methods for acoustic sub-word based systems are discussed in [12], [13]. In this paper, we use employ a first order Markov model as the statistical lexicon [12] for building SI systems.

While speaker independent (SI) speech recognition systems can show impressive performance, speaker trained or speaker dependent (SD) systems can provide an word error rate (WER) a factor of two to three lower than an SI system if both systems use the same amount of training data. Hence the major rationale for investigating speaker adaptive (SA) systems is that they promise to produce a final system that has desirable SD-like properties but requires only a small fraction of the speaker-specific training data needed to build a full SD system. Adaptation can significantly improve the WER for outlier speakers such as non-natives or others not well represented in the SI training set. The earliest approaches to speaker adaptive systems were are MAP based ([15],[16]). But the MAP based methods require considerable amount of adaptation data from the test speaker.

Regression based ([17] - [19]) and transformation based ([20] - [21]) approaches to speaker adaptation were, to some extent, successful in reducing the amount of adaptation data and yet providing good performances. More recently, rapid adaptation techniques using the eigenvoices ([22] - [25]) have been used widely. the task of speaker and channel adaptation. The approach constrains the adapted model to be a linear combination of a small number of basis vectors obtained offline from a set of training speakers, This, therefore, reduces the number of parameters to be estimated from the adaptation data. The eigenvoice basis vectors are orthogonal to each other and are guaranteed to represent the most important components of variation among the training speakers. Modifications of the basic method like the splitting of eigenvoices into sub-spaces [26] and kernel based high dimensional mapping of the eigenvoices ([27], [28]) help to further reduce the amount of adaptation data.

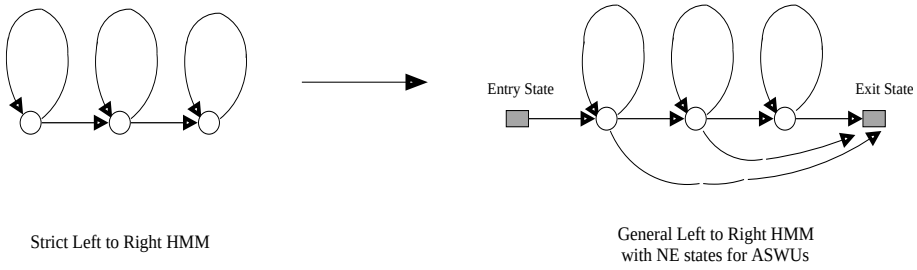

Fig. 1. Illustration of Strict L-R HMM used for WWU and the General L-R HMM used for the ASWU

In this paper, we use the basic eigenvoice method [24] for adapting the Acoustic sub-word models. Both supervised and unsupervised adaptation has been explored. The paper is organized as follows. Section II explains the iterative HMM based clustering method for obtaining the sub-word models. The procedure for eigenvoice adaptation of the sub-word HMMs is discussed in III. The experiments and results are reported in IV followed by a discussion in V. We finally conclude in VI.

II. ASWU MODELING

The ASWUs are modeled by general left to right HMMs. The difference between a traditional strict L-R HMM and a general L-R HMM is illustrated in Fig.(1). By using this model for the ASWUs, the modeling accuracy is improved as compared to a strict L-R HMM.

A novel technique for training ASWU models (Fig. 2) is explained now. Compared with the techniques in literature, initial clustering using spatial interpolation and iterative HMM training are the two contrasting steps in this model building procedure. The important steps involved are -

A. ML Segmentation

Speech signals of different speakers are analyzed in short time windows to obtain short term spectral representations i.e. MFCCs. The feature vector sequences are segmented using the Maximum Likelihood (ML) criterion based on a fixed number of segments per sec. The ML segmentation algorithm [8] is used for constrained duration segmentation, i.e., although the number of segments in a speech utterance is to be found analytically, we use, for simplicity, a fixed segmentation rate of 10 segments per sec. The length of a segment varies from $lmin$ frames per segment to $lmax$ frames per segment. The size of speaker independent ASWU inventory determines the spectro-temporal resolution of the acoustic segment space. For our work, it is chosen to be 64.

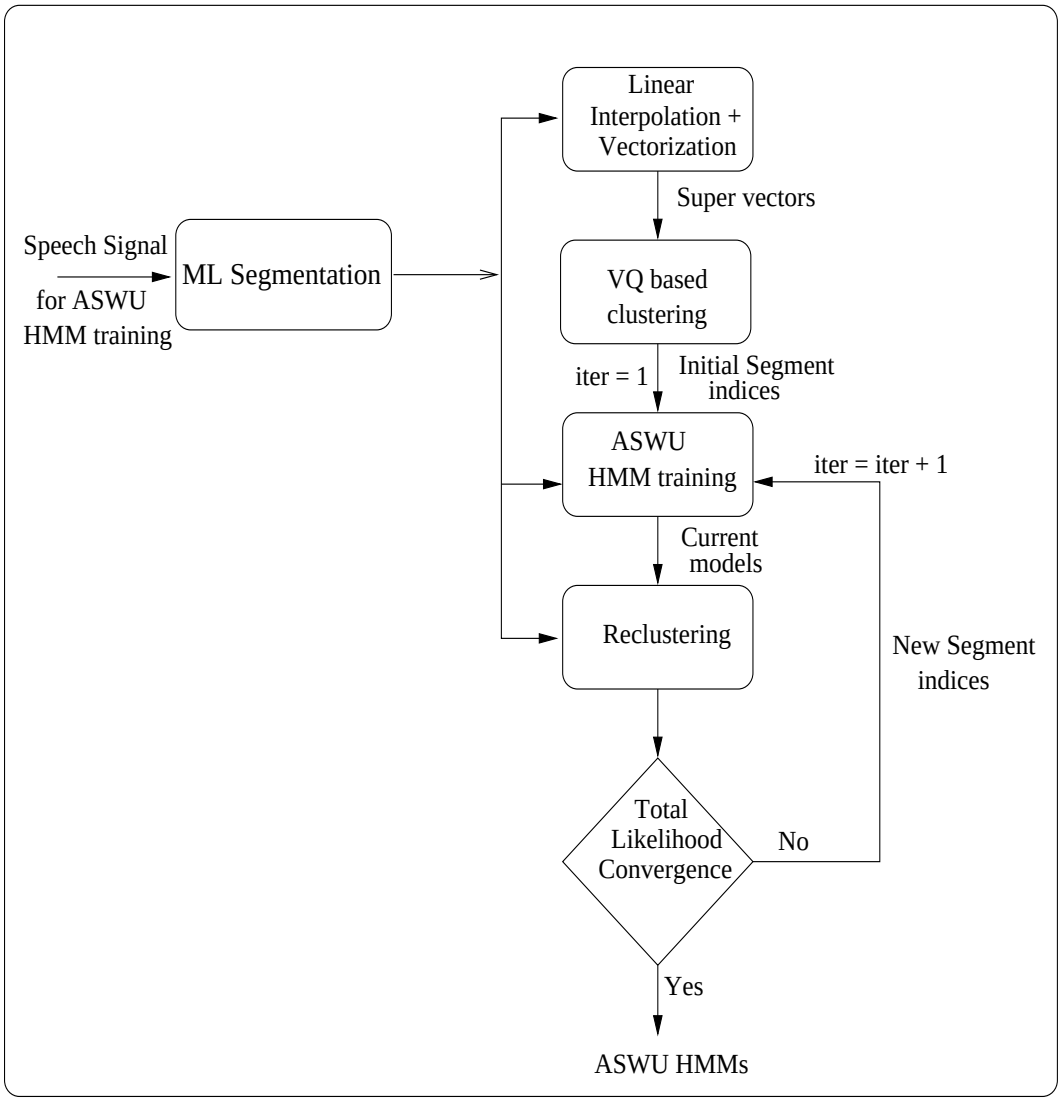

Fig. 2. Block Schematic showing Sub-Word HMM training

B. Iterative Segment Clustering and Modeling

Step 1. Interpolation - All the segments used for training the sub-word models are linearly interpolated in the feature space to make them equally sized ($lmax$ vectors per segment). This interpolation can be interpreted as a spatial interpolation of the feature vectors in a high dimensional feature space. For example, a segment of size 4 is interpolated to size 6 as shown in Fig.(3). This process is continued until segment size becomes $lmax$.

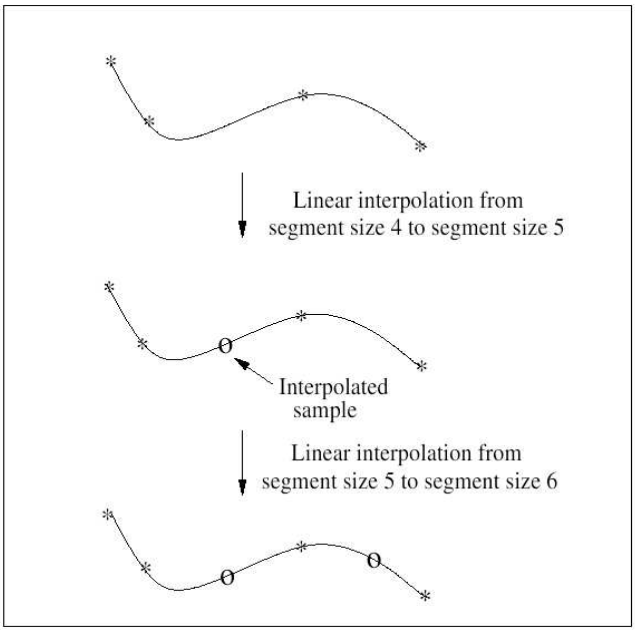

Fig. 3. Illustration of the Spatial Interpolation

Step 2. Vectorization - Each interpolated segment is now vectorized to form a single long vector, i.e. the $lmax$ vectors, each of dimension $D \times 1$, representing an interpolated segment is now put in a single column of size $lmax \times D$. We call this long vector a super vector. Thus, each segment is now represented by a super vector.

Step 3. Initialization - A Vector Quantizer is designed on the super vectors. The VQ codebook size is the size of the ASWU inventory - 64. Once the code-book is obtained, all the super vectors are clustered to obtain their sub-word labels. This step provides an initial clustering for building the sub-word HMMs. Set $iter = 1$.

Step 4. Model Building - All the training segments (without interpolation) belonging to the same sub-word index (for the current iteration) are used for building its sub-word HMM. We get 64 ASWU HMMs at the end of this step.

Step 5. Reclustering - The training segments are reclustered using the sub-word HMMs. i.e., each training segment is classified as one of the sub-word indices using the ASWU HMMs of the current

iteration. Thus we get a new set of segment indices for the training segments The total likelihood of the segments is also calculated at the end of this step. The total likelihood P_{total} is defined as

$$P_{total} = P_1 \times P_2 \times P_3 ..... \times P_S \tag{1}$$

where S denotes the total number of training segments and P_i, $i = 1, 2, ..., S$, denotes the likelihood of the i^{th} segment given its sub-word model.

Step 6. Check for convergence - We define the quantity Δ as

$$\Delta = \frac{P_{total}^{iter} - P_{total}^{(iter-1)}}{P_{total}^{(iter-1)}} \tag{2}$$

If $\Delta < \Delta_{th}$ we stop the reclustering process. The models at the end of Step 4 for the current iteration gives the sub-word HMMs. Else, using the new set of sub-word indices obtained at end of Step 5 for the current iteration, goto Step 4 for building the new sub-word HMMs and set $iter = iter + 1$.

This method of reclustering resembles the unsupervised K-means clustering. The initialization procedure for building the sub-word HMMs itself provides a good segment labeling scheme. But, because sub-word units have to built in speaker independent manner, we make use of the iterative procedure for obtaining the ASWU HMMs. We find, in our experiments, that the total likelihood P_{total} is monotonically increasing with iterations, thereby, making the segment labeling process better and better with each iteration.

For recognition, it is necessary to build a lexicon (dictionary) for all words in the vocabulary.

C. Lexicon Building

The word utterances from the training speakers are provided to the Viterbi decoding block in the null-grammar mode and output sub-word sequence is the requisite lexicon. With multiple training samples available for a word, the lexical decoding is performed for each training sample and a representation sequence/s is used for building the lexicon of that word [12].

In our experiments, from the training data, we observe the following -

1) For a particular word, the sub-word sequences for different utterances from the same speaker do not vary much.
2) For the same word, the sub-word sequences for utterances from different speakers varies quite a lot.

As an example, we provide the sub-word sequences for the vocabulary word - “One” for two different speakers.

Spoken by	Sub-word Sequence
Speaker 1, utterance 1	56 58 6 9 17 14 56
Speaker 1, utterance 2	56 58 5 9 17 14 56
Speaker 2, utterance 1	49 57 41 45 15 58 50 49

This sort of lexical difference is found for all words in the vocabulary.

We do not focus much on the lexicon building issue, as it is a challenging problem by itself. For our work, we pursue either a statistical lexicon for the speaker independent mode of operation and a speaker dependent(SD) deterministic lexicon.

SD lexicon - corresponds to the one obtained from the first utterance (labeled) of all the vocabulary words from the test speaker; the adapted models are used to test the subsequent two utterances of the test speaker. We briefly describe the statistical lexicon building process now.

Statistical lexicon - A first order Markov chain is used as a lexicon for each word in the vocabulary [12]. The states of the Markov chain correspond to the ASWU indices, and therefore, the Markov chain has 64 different states. The initial and transition probabilities fully determine the Markov chain and these are learnt using the different sub-word sequences of that word from the training speakers.

The use of this type of a statistical lexicon is better than the employment of multiple deterministic lexicons per word [12].

We are now ready to discuss the adaptation algorithms for the IWR system based on acoustic sub-words.

III. Adaptation Based on sub-word Eigenvoices

Eigenvoice method of rapid adaptation [24], is employed to adapt the sub-word HMMs. The important issues discussed here are

1) Determining the sub-word eigenvoices.
2) Model Adaptation with statistical word lexicon.
3) Model Adaptation with Speaker Dependent lexicon.

A. Determining the sub-word eigenvoices

The presence or absence as well as the number of occurrences of ASWUs are very much speaker dependent. Hence, the eigenvoices are determined in a sub-word specific manner , i.e. eigen analysis is

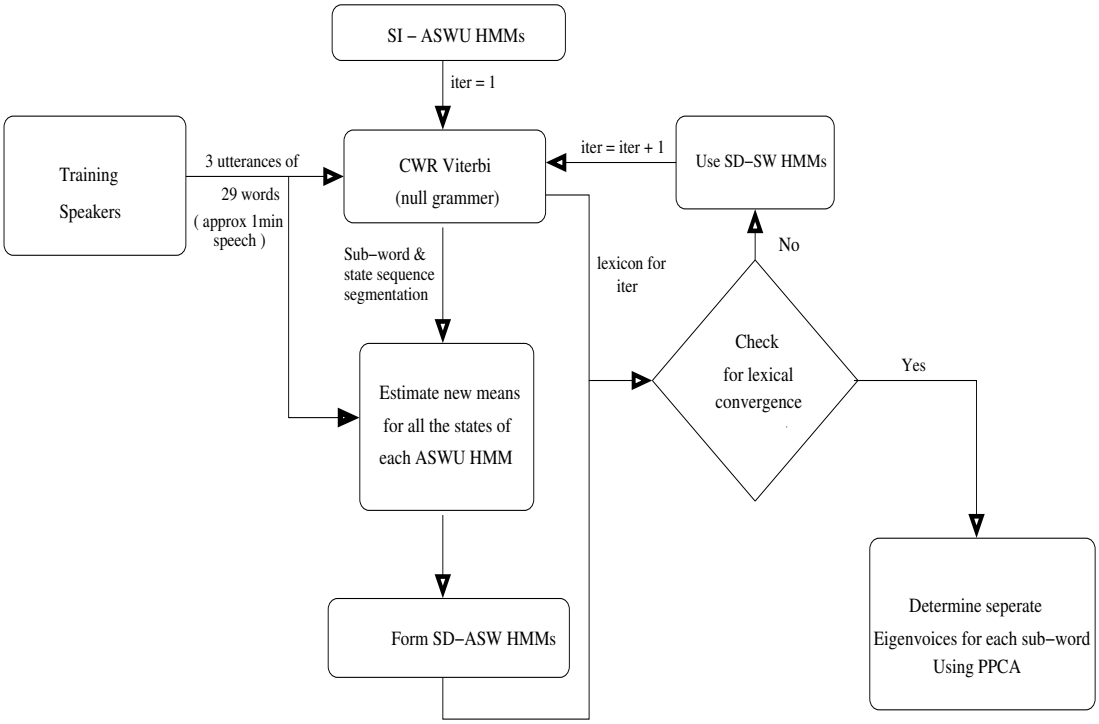

Fig. 4. Algorithm for determining sub-word eigenvoices

done separately on the model parameter space of each sub-word unit. This in contrast to eigenvoice techniques in literature ([24]), where eigenvoices are determined on very high dimensional space containing the parameters of all sub-word units. Also, since the number of training speakers is very less (only 20), this sort of eigen analysis is more statistically stable.

The algorithm for determining the eigenvoices from the training speakers is illustrated in Fig.(4).

Step 1. - For each training speaker, all utterances of the vocabulary words (approx 1 min of speech for our data base of vocabulary size 29 and with three utterances per speaker) are concatenated to form a single long speech signal.

Step 2. - The SI ASWU HMMs are provided to the CWR-Viterbi stage. Set $iter = 1$.

Step 3. - Using the models of the current iteration and the training speech of Step 1, the CWR Viterbi algorithm is operated in the null grammar mode to obtain sub-word decoding, sub-word segmentation and the best state sequence segmentation within each sub-word.

Step 4. - The ASWU segmentation and the state sequence segmentation from Step 3 is used along with the training data to estimate SD means for all state observation densities of ASWU HMMs. If a state of a sub-word or a sub-word itself has not appeared in the training speech of that speaker, its SI mean is retained in place of the SD estimate.

Step 5. - The SD means estimated from the previous step is used along with the SI covariances and state transition probabilities[1] to form a set of SD ASWU HMMs for this training speaker.

Step 6. - Check for convergence. We define the quantity Δ as

$$\Delta = \mathbf{d}\Big[lexicon(iter)\ ,\ lexicon(iter-1)\Big] \tag{3}$$

where $\mathbf{d}$ denotes the Levenstein distance (also called the string edit distance [29]) and the string here denotes the sub-word sequence for training speech of Step 1 for the particular training speaker. In calculating Levenstein distance, the insertion, deletion and substitution cost are set to 1. If $\Delta < \Delta_{th}$, we stop the mean estimation iterations. The SD means for this training speaker at the end of Step 3 is input to Step 7. Else, goto Step 3 with the adapted sub-word HMMs of Step 5. Set $iter = iter + 1$.

Step 7. - Steps 1 - 6 are repeated for all the speakers in the training corpus. Since some sub-words are specific to some training speakers, it is improper to build a single long supervector consisting of all the sub-word state HMMs and do a single eigen analysis for this super-vector. Instead, using all the training speaker's SD means (for all the states) of a particular sub-word, we obtain the eigenvoices specific to that sub-word. The number of distinct eigenvoices obtained for each sub-word, varies depending on the number of training speakers contributing the particular sub-word; when a speaker does not contribute it will be an SI model parameter, resulting in a lower rank representation.

Thus, we get a set of eigenvoices for each sub-word; the space spanned by these eigenvoice vectors for a particular ASWU corresponds to the speaker variation space for the model parameters of that sub-word. Because the eigenvoices are sub-word dependent, we call them **sub-word eigenvoices**. These sub-word eigenvoices are used in the adaptation task.

[1]This is an important design choice, since we adapt only the mean vectors and it is assumed that there is only minimal data in rapid adaptation, so that reliable estimates of covariances cannot be obtained from the adaptation data

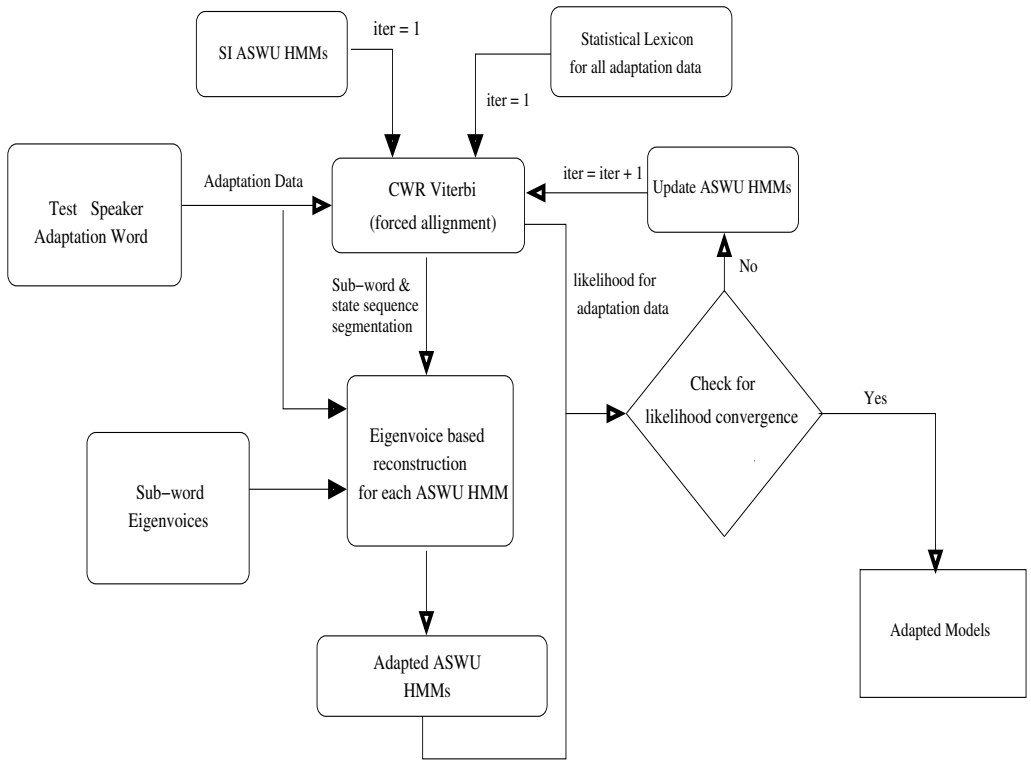

Fig. 5. Algorithm for supervised adaptation with statistical word lexicon

B. Model Adaptation with statistical word lexicon

The statistical lexicon refers to the first order Markov lexicon explained in Sec.(II-C). We use a supervised adaptation procedure here as shown in Fig.5. The supervised adaptation refers to the use of a known vocabulary word, for which the statistical lexicon has already been obtained from the training data. The important steps are

Step 1. - For a test speaker, his supervised adaptation data is provided for the CWR Viterbi algorithm. The SI ASWU HMMs and the statical lexicon for the words in the adaptation data are also provided to the connected decoding stage. Set $iter = 1$.

Step 2. - The CWR Viterbi algorithm is operated in the forced alignment mode to determine the likelihood of the adaptation data w.r.t. the sub-word models of the current iteration. i.e.,

$$P_{iter} = P(\mathbf{O}^{adaptation}|\lambda_{ASWU}^{iter}) \tag{4}$$

The lexical decoding, sub-word segmentation and the best state sequence segmentation within each

sub-word is also obtained as a byproduct of the CWR-Viterbi algorithm.

Step 3. - The best ASWU segmentation and the state sequence segmentation obtained from Step 2 is used along with the adaptation data and the sub-word Eigenvoices to obtain the adapted means for all the state observation densities of ASWU HMMs. If a state of a sub-word or a sub-word itself has not appeared in the adaptation data, its SI mean is left unadapted.

Step 4. - The adapted means from the previous step is used along the SI covariances and transition probabilities to form a set of adapted ASWU HMMs for the test speaker.

Step 5. - Check for convergence. We define the quantity δ as

$$\delta = abs\left[\frac{P_{iter} - P_{iter-1}}{P_{iter-1}}\right] \tag{5}$$

If $\delta < \delta_{th}$, we stop the mean adaptation process. The adapted models at the end of Step 3 is input to Step 6. Else, goto Step 2 with the adapted sub-word HMMs of Step 4 for the current iteration. Set $iter = iter + 1$.

Step 6. - The adapted ASWU models are used for the recognition of unseen word utterances from the test speaker.

The recognition stage is done by using the statistical word lexicons of each word in the vocabulary learnt from the training speakers; the algorithm adapts only the ASWU HMMs from the adaptation data and does not change the word lexicons. Thus, for any test speaker, data for all vocabulary words is not a must for this algorithm. A portion of the vocabulary, if provided for adaptation, will suffice. However, the adaptation words are pre-determined (supervised) whose lexicon is already learnt from the training speakers.

C. Model Adaptation followed by SD lexicon building

The block schematic for this algorithm is shown in 6. The following are the important steps.

Step 1. - For a test speaker, the unsupervised adaptation data and the set of SI ASWU HMMs is provided to the CWR Viterbi algorithm. Set $iter = 1$.

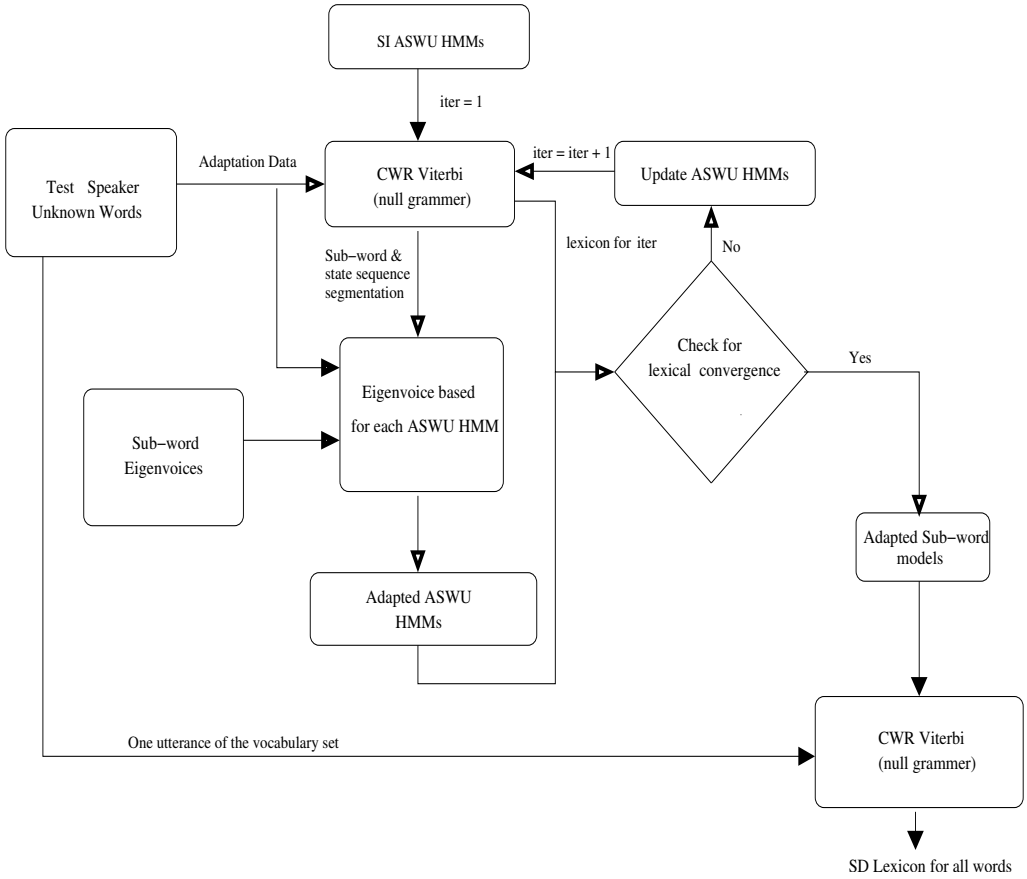

Fig. 6. Algorithm for unsupervised adaptation with speaker dependent lexicon

Step 2. - The CWR Viterbi algorithm in the null-grammar mode gives the lexical decoding of the adaptation data.

Step 3. - The best ASWU segmentation and the state sequence segmentation obtained from Step 2 is used along with the adaptation data and the sub-word Eigenvoices to obtain the adapted means for all the state observation densities of ASWU HMMs. Sub-words not seen in the adaptation data are left unadapted.

Step 4. - The adapted means estimated from Step-4 is used along with the SI covariances and transition probabilities to form a set of adapted sub-word HMMs for this test speaker.

Step 5. - Check for lexical convergence. We define the quantity Δ as in Eq.(3). If $\Delta < \Delta_{th}$, we stop

the adaptation iterations. The adapted models at the end of Step 4 is input to Step 6. Else, goto Step 2 with the adapted sub-word HMMs of Step 4 for the current iteration. Set $iter = iter + 1$.

Step 6. - Using the adapted ASWU models and one utterance each of all vocabulary words from the test speaker, the SD lexicon for all the vocabulary words is obtained.

Step 7. - The SD lexicon and the adapted ASWU HMMs are used for recognizing the unseen words from the test speaker.

It may be noted that in the unsupervised mode, since we have not used any lexical information from the training speakers, out of vocabulary words can be also be used as a adaptation data. This is quite interesting as it is different to the conventional adaptation techniques where, such a flexibility is not present.

IV. Experiments and Results

A. Data Base

Experiments are conducted on BPL database. The data base consist of Indian Accented English Isolated Words in microphone channel(16 KHz sampling) spoken by 32 female and 38 male adult speakers. The vocabulary for isolated words is 29 voice dialer application words, which includes the ten digits and general call dialing application words. Three repetitions of each word is recorded per speaker. The words spoken are properly end-pointed using an automatic Voice-Activity Detection (VAD) algorithm and the beginning and end points are manually checked for errors.

B. ASWU Modeling

For the purpose building the ASWU models, these isolated words of the training speakers are concatenated, i.e., the 29 words of one repetition of a speaker are concatenated to form an approx 20s long speech data. This is repeated for all utterances of a speaker and for all the speakers in the training corpus. Thus we have approximately $20 trainingspeakers * 3 repetitions * 20s = 20 min$ of speech for obtaining the Speaker Independent ASWU models.

The speech signals are parameterized into 13-dimensional MFCC feature vector sequences (12 cepstral values + log energy). To remove the channel effects from the speech utterances, the MFCC vectors are mean subtracted (Cepstral Mean Subtraction) and log energies are normalized. Now the feature

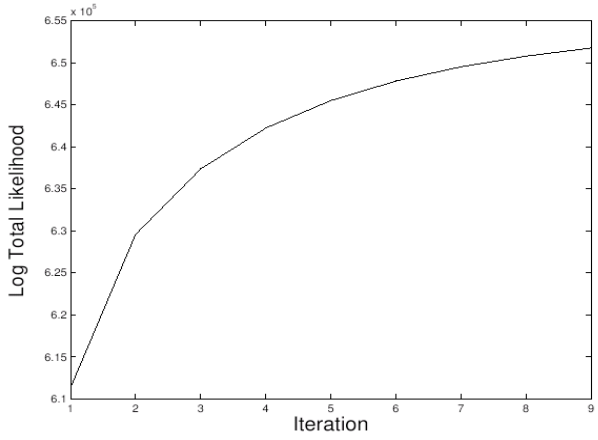

Fig. 7. Plot of total likelihood for the training data versus iterations

sequences for all the speech sentences are segmented using the ML criterion. As mentioned in Sec.II, the segmentation is done at a rate of 10 segments per second. The minimum segment duration $lmin$ was chosen to be 8 and the maximum segment length $lmax$ is set to be 12, so that, on an average, there are 10 segments for every second of speech.

The segments are interpolated and clustered into 64 size codebook. This provides an initialization for building the 64 sub-word HMMs. Each ASWU is modeled using a 3-state CDHMM having single Gaussian distribution as the state observation density. Once the initial HMMs are built, we do a reclustering process as explained in (II-B). The total likelihood (1) for the training segments is found to monotonically increase with iterations Fig.7). This shows that the modeling as well as the labeling process improves with iterations.

C. Adaptation Experiments

20 speakers are used for training and the remaining 50 speakers are used for testing the adaptation performances. Speech data for training the ASWU models consist of 3 utterances each of 29 words from the 20 training speakers. The procedure for training the ASWU models, as explained in Sec. II, is used to obtain a set of ASWU HMMs. Using the ASWU models, a first order Markov model representing the lexicon of each word is trained. This type of lexicon is used for system configurations (**1**) and (**2**) in Fig.8, which shows the various configurations in which the experiments have been done.

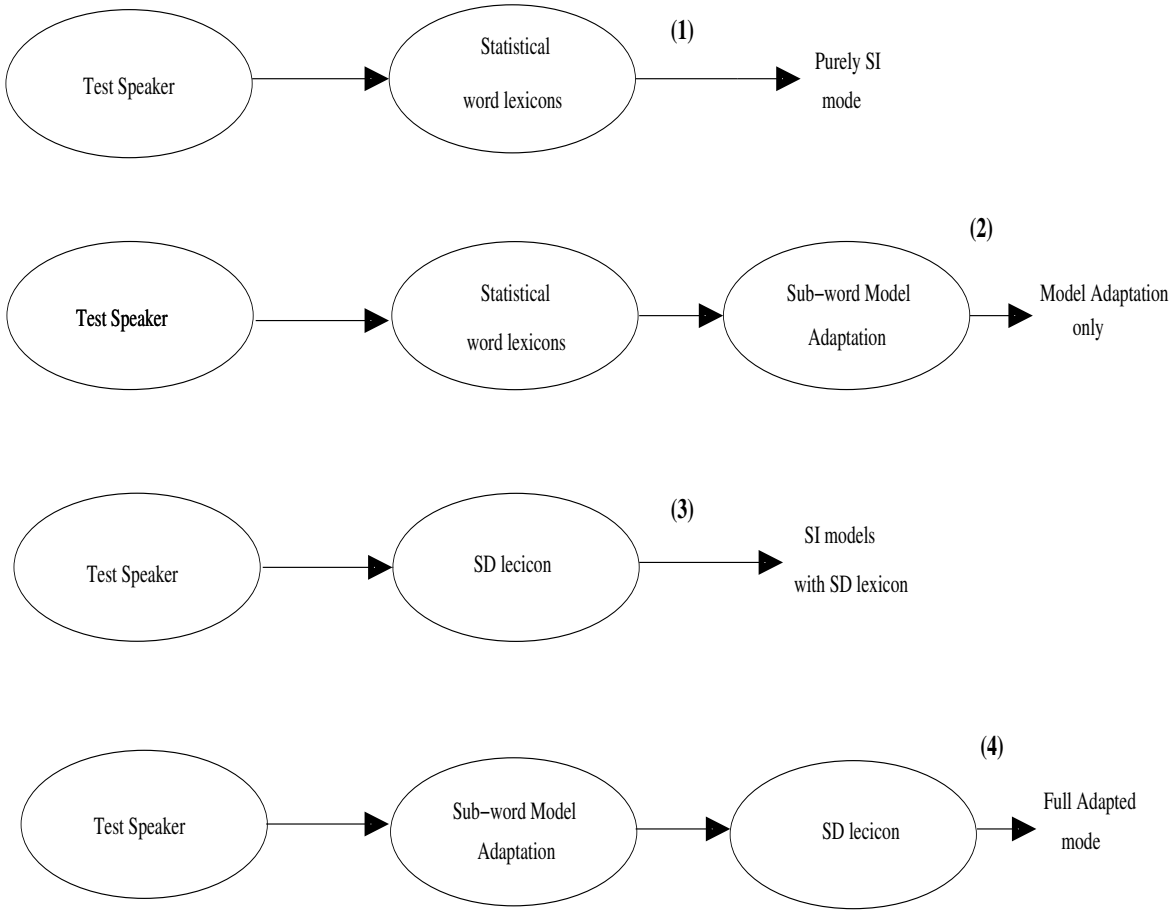

Fig. 8. Various System configuration employed in the experiments

In all four experiments, we report the recognition performance for the 50 test speakers and 29 words, using the last two repetitions (2900 words). We have discussed two types of adaption, supervised and unsupervised. These two cases are referred to as configurations **2** and **4** respectively. The topologies **1** and **3** act as benchmark for **2** and **4** respectively.

1) **Pure SI mode** - This is the mode of recognition system, that does not use any data from the test speaker. The SI ASWU models and the statistical lexicon for the words built during the training stage, are used for recognition on the unseen test speaker's data. For all the adaptation experiments, this forms the base-line performance on which we improve upon.
2) **Model Adaptation only** - The supervised adaptation data along with the statistical lexicon is used for adapting the sub-word HMMs (Sec.III-B).
3) **SI models with SD lexicon** - The first utterance of a word from the test speaker is used to

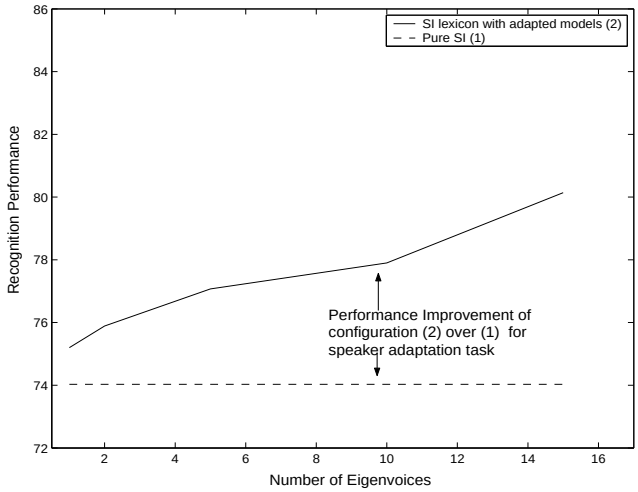

Fig. 9. Adaptation Performance with statistical lexicon

build his SD lexicon for that word. Recognition is performed using the SI ASWU HMMs and SD lexicon (no adaptation).

4) **Adapted models with SD lexicon** - Unsupervised adaptation data from the test speaker is used to adapt the sub-word models, followed by lexicon building based on the adapted models (Sec.III-C).

Intuitively, configuration (**4**) should perform the best. *The main intention, in our adaptation experiments here, is to show the performance improvement of configuration (**2**) over (**1**) and that of configuration (**4**) over (**3**)*. The other performance improvement of (**3**) over (**1**) will denote the importance of lexical adaptation.

D. Adaptation with statistical lexicon

Here, we compare the results of experiments (**1**) and (**2**). The adaptation data for experiment-(**2**) consist of the first utterance of all the words by the test speaker(supervised). The following figure (Fig. 9) shows the adaptation performance, for the speaker adaptation task, as the number of eigenvoices is varied. The actual recognition performance obtained for these experiments is shown in Table-IV-D

Fig.(9) shows that higher the number of eigenvoices, the better the system performs, but at a higher computational complexity. Also, it is interesting to note the increase in performance after $K = 10$, showing that even 10 eigenvoices are not sufficient to model the speaker variation space of the 20 training speakers.

The row $K = max$ in Table(IV-D), here denotes the maximum possible - equal to one less than the

TABLE I

RECOGNITION PERFORMANCE FOR ADAPTATION WITH STATISTICAL LEXICON - SPEAKER ADAPTATION TASK.

Number of eigenvoices	Pure SI Performance - (**1**)	SI lexicon + adapted models - (**2**)
K = 1	74.03%	75.20%
K = 2	74.03%	76.21%
K = 5	74.03%	77.07%
K = 10	74.03%	77.90%
K = max	74.03%	80.14%

rank of each sub-word's eigenvoice matrix (matrix with the eigenvoices as columns). Since different sub-word eigenvoice spaces are of different dimensions, (because of lack of training data for some sub-words from all the training speakers) the maximum number of eigenvoices possible for different sub-words is different.

As can be seen, the adaptation of models alone gives a good improvement in performance. **Even without adapting the lexicon, the model adaptation provides a reduction of upto 23.50% in the word error rate.** It is worth remembering here that this improvement in performance is provided by about 20s of supervised adaptation data.

E. Adaptation followed by SD lexicon building

We now compare the results of experiments for system topologies (**3**) and (**4**). The adaptation data for these experiments consist of the first utterance of all the words by the test speaker (unsupervised). The interesting difference between Fig.(10) and Fig.(9) is that, in former one, there is not much improvement in performance after $K = 10$. Thus, we may say that 10 eigenvoices are sufficient for model adaptation when we use a speaker dependent lexicon.

The first observation here is the improvement of configuration (**3)** over (**1)**. This shows the importance of having a good lexicon for the words in the vocabulary when the IWR system is ASWU based. *The other observation, important from the point of view of speaker adaptation, is the considerable performance improvement (around **44.53%** reduction in word error rate for $K = max$) obtained by adapting the models and then building the test speaker SD lexicon.*

It is also worth noting that, experiment-3 performance is better than that of experiment-2; i.e, with no adaptation, but SD lexicon we can get good performance. But this is useful only for small size

TABLE II

ADAPTATION PERFORMANCE USING A SPEAKER DEPENDENT LEXICON.

Number of eigenvoices	SD lexicon + SI models - (**3**)	Full adapted - (**4**)
K = 1	82.03%	84.45%
K = 2	82.03%	86.28%
K = 5	82.03%	87.14%
K = 10	82.03%	89.48%
K = max	82.03%	90.03%

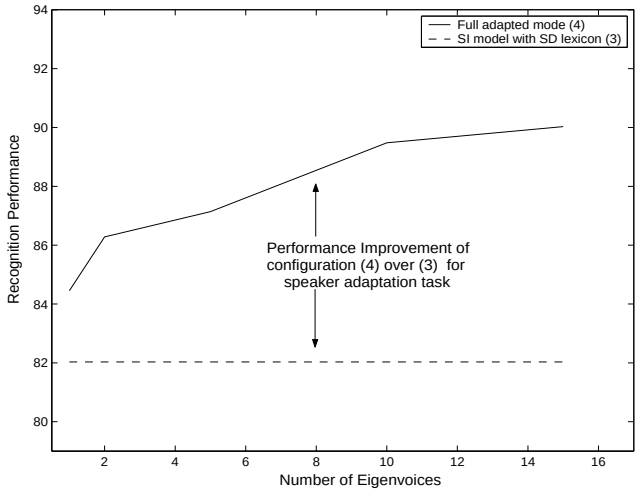

Fig. 10. Adaptation Performance with SD lexicon for speaker adaptation task.

vocabularies and not useful for large vocabulary. It may be noted that **the unsupervised adaptation mode does not show much difference in performance for speaker adaptation tasks**. This is probably because after full adaptation, all the characteristics of the training speakers are adapted.

Fig.(11) shows a comparison of adaptation performance of all the IWR system configurations discussed in this paper. The WWU based ASR system's results is also included in this figure so as to obtain a comparison between the whole word and sub-word performances. As can be seen from the figure, the WWU based systems are better than the ASWU based system (either with SI lexicon or SD lexicon). This is because, the WWUs can explicitly incorporate the effects of inter-phonemic context dependence and coarticulation in their word models. But, the full adapted mode performs better than the SI WWU

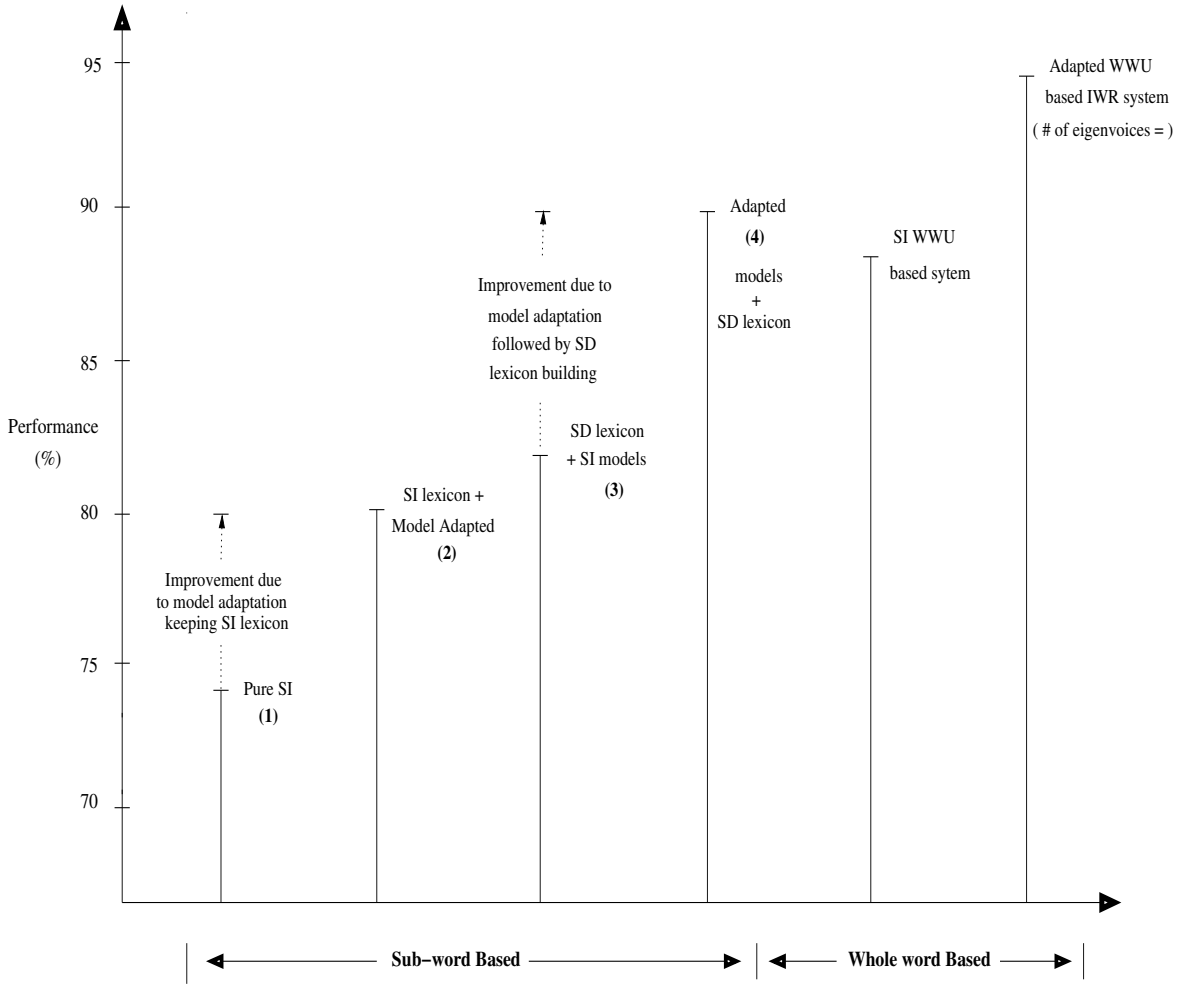

Fig. 11. Performance comparison for the different ASWU based experiments with maximum number of eigenvoices and the WWU based system for the speaker adaptation task.

(no adaptation) based IWR system. The figure clearly illustrates the fact that adapting models by the eigenvoice algorithm can consistently improve the recognition performance for both cases (with statistical lexicon (**1**) and with speaker dependent lexicon(**3**). This figure also emphasizes the need for lexical adaptation (improvement in performance of lexical adaptation (comparison of topology (**1**) and (**3**)) is slight better than model adaptation performance improvement with a statistical lexicon (comparison of topology (**1**) and (**2**)).

V. Discussion

The IWR configuration-(**1**), which uses supervised model adaptation and SD lexicon based on the adapted models, is used for some interesting experiments. The two important questions we try to answer

here are :-

1) Can a similar performance be achieved with a reduced amount of adaptation data ?
2) If answer to 1 is yes, then can we still improve the performance from the amount of adaptation data currently used ?

These experiments give clear insight into the eigenvoice adaptation algorithm. Keeping the liberty of forming the SD lexicon of all the vocabulary words after adapting the ASWU models, and keeping the number of eigenvoices, K at the maximum, we obtain the following.

TABLE III

RECOGNITION PERFORMANCE FOR REDUCED ADAPTATION DATA FOR ADAPTATION EMPLOYING SD LEXICON

Amount of partial adaptation data	Performance for partial data
6 words (4s)	84.59%
10 words (7s)	86.21%
15 words (10s)	88.38%
20 words (13s)	88.69%
Full Data (20s)	90.03%

The full data adaptation here denotes the case when the all first utterance are used as the adaptation data. We can see a saturation effect with the performance of 29 words with adaptation is quite comparable to that with 29 words. Thus, the additional amount of data does not increase the performance much. Also, the drastic improvement in performance from 6 words adaptation data to that when 20 words are provided as adaptation data is due to the fact that, out of 64 ASWUs, only around 25-30 different sub-words are seen in the adaptation data for the former, whereas about 40-45 sub-words are seen in the adaptation data of the later.

Next, we try another experiment by limiting the adaptation data to have only n or less utterances for each sub-word. This is done by first forming the lexicon for 20s of full adaptation data and then selecting, in each iteration, the portion of data corresponding to the first n utterances of all the sub-words and their corresponding state sequence segmentations. We have experimented with $n = 1$ and $n = 2$.

The table above clearly shows the advantage as well as disadvantage of the Eigenvoice based adaptation algorithm. The advantage is that the algorithm is really able to give good performance improvements even with two utterances per sub-word (7s of adaptation data). This is very good for an LVCSR task

TABLE IV

RECOGNITION PERFORMANCE ADAPTATION DATA WITH FEW NUMBER OF UTTERANCE PER SUB-WORD

No: of utterances per ASWU	For partial data	Full data adaptation (20s)
1 utterance per ASWU (4s)	87.24%	90.03%
2 utterances per ASWU (7s)	89.86%	90.03%
2 utterances per ASWU + cordless channel	89.52%	90.62%

where around 15 - 20s of speech will contain almost all the sub-words atleast twice. Thus, we have shown that the **Eigenvoice technique fares well as a rapid adaptation algorithm**.

Having found that the adaptation data provided for some frequently occurring sub-words is not fully utilized by the eigenvoice algorithm, we try to find if it is possible to improve the performance when first utterance of the 29 words is provided data (full adaptation data - 20s). As sufficient data is available for the individual states of ASWU HMMs, the eigenvoices are formed for each state of each sub-word HMM. Therefore, we have $64 * 3 = 192$ set state sub-word eigenvoices and each is of dimension 13. The state sequence segmentation obtained from the CWR-Viterbi algorithm, helps to obtain the adaptation data for each state of each sub-word, which is utilized to adapt that state's mean of that sub-word HMM.

TABLE V

STATE DEPENDENT EIGENVOICE ADAPTATION WITH FULL FIRST UTTERANCE AS ADAPTATION DATA.

No: of eigenvoices	Performance
K = 5	89.66%
K = 12	90.48%
K = 5 and partial sample mean	90.38%
K = 12 and partial sample mean	91.52%

where, partial denotes the case when there are more than a predefined number of data vectors a particular sub-word HMM's state, we simply take the sample mean of the data vectors having that state index., i.e., if the number of data vectors belonging to the same sub-word state HMM exceeds a preset threshold, a sample mean of these vectors is the SD mean for that ASWU HMM's state, else the state

dependent eigenvoice adaptation of the mean is performed. The threshold for these experiments was set as 13. In the experiments, we find that about one-third of state means get sample mean estimates and another one-third gets adapted by eigenvoice based algorithm and last one-third remain un-adapted. The table above shows that, whenever there is more amount of adaptation data one should go for more sophisticated algorithms like MLLR or MAP based adaptation schemes, as eigenvoice adaption technique cannot improve the performance much in such a case.

VI. SUMMARY

We have addressed the task of building automatic tokens for the task of speech recognition. The segmentation of speech signal is done using an acoustic criterion (ML-segmentation). We believe that consistency and discriminability among the speech units are the key issues for the success of speech systems. We develop a new procedure for clustering and modeling the segments using sub-word HMMs. Because of the liberty in choosing the labels for the sub-word units, we do an iterative reclustering and modeling of the segments.

We have also applied the eigenvoice adaptation technique to a sub-word based IWR system. We conclude from our experiments that acoustic units provide a great deal of flexibility and freedom for adaptation, as the units are totally data driven. ASWU HMMs, when adapted, can provide performances compared to WWU based systems. The results are interesting and really encourage us to pursue adaptation of ASWU models in more tougher and challenging tasks like LVCSR and channel adaptation tasks.

REFERENCES

[1] J.G. Wilpon, B.H. Juang and L.R. Rabiner, “ An Investigation on the Use of Acoustic Sub-Word Units for Automatic Speech Recognition,” *Proc. of ICASSP*, pp. 821-824, Apr. 1987.

[2] Maen M. Artimy, William Robertson and William J. Phillips, “ Automatic Detection of Acoustic Sub-word Boundaries for Single Digit Recognition” *Proc. of IEEE Canadian Conference on Electical and Computer Engineering*, pp. 751-754, Alberta, Canada, May 1999.

[3] V.Ralph Algaz and Kathy L. Brown, “Automatic Speech Recognition using Acoustic Sub-words and No Time Allignment, ” *Proc. of ICASSP*, pp. 465-468, Apr. 1988.

[4] V.Ralph Algazi, Kathy L. Brown, Michael J. Ready, David H. Irvine, Christie L. Cadwell and Sang Chung “Transform Representation of the Spectra of Acoustic Speech Segments with Applications-I: General Approach and Application to Speech Recognition, ” *IEEE Trans. Speech and Audio Proc.*, vol.1, pp. 180-195, Apr. 1993.

[5] Kathy L. Brown and V.Ralph Algani, “Charecterization of Spectral Transitions with Applications to Acoustic Sub-word Segmentation and Automatic Speech Recognition, ” *Proc. of ICASSP*, pp. 104-107, May 1989.

[6] Regine Andre-Obrecht, “A New Statistical Approach for the Automatic Segmentation of Continuous Speech Signals, ” *IEEE Trans. Speech and Audio Proc.*, vol. 36, pp. Jan 1988, pp. 29-40.

[7] C.H. Lee, F.K.Soong, B.H.Juang,“ A Segment Model Based Approach to Speech Recognition”, *Proc. ICASSP*, pp. 501-904, April 1988

[8] T.Svendsen, F.K Soong: ”On the automatic segmentation of speech signals”, *Proc. ICASSP*, pp. 77-80, April 1987

[9] T.Svendsen, K.K. PaIiwal, E.Harborg, P.0. Husoy, “An Improved Sub-Word Based Speech Recognizer”, *Proc. ICASSP*, pp. 108-111, May 1989

[10] A.K.V. Sai Jayram, V.Ramasubramanian, T.V. Sreenivas, “Language Identification Using Parallel Sub-word Recognition”, *Proc. ICASSP*, pp. 729-732, May 1990.

[11] Trym Holter and Torbjorn Svendsen, “Combined Optimisation of Baseforms and Model Parameters in Speech Recognition Based on Acoustic Subword Units”, *IEEE Workshop on Automatic Speech Recognition and Understanding* , pp. 199-206, Dec. 1997.

[12] K.K. Paliwal, “Lexicon Building Methods for an Acoustic Sub-word Based Speech Recognizer ”, *Proc. ICASSP*, pp. 32-35, May 2003.

[13] T. Holter and T. Svendsen, “A comparison of lexicon-building methods for subword-based speech recognisers,” *in Proc. IEEE Region 10 Conf. on Digital Signal Proc. (TENCON)*, Perth, Australia, pp. 102-106, Nov. 1996.

[14] Lawrence Rabiner and Biing-Hwang Juang, “Fundementals of Speech Recognition”, Prentice Hall PTR 1993.

[15] Chin-Hui Lee and Jean-Lue Gauvain, “Speaker Adaptation Based on MAP Estimation of HMM Parameters”, *Proc. of ICASSP* pp-558-561, May 1993.

[16] Jean-Lue Gauvain and Chin-Hui Lee, “Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains”, *IEEE Trans. Speech and Audio Proc.*, vol. 2, No. 2, April. 1994, pp 291-298.

[17] Stephen Cox, “Predictive Speaker Adaptation in speech recognition”, *Computer Speech and Language*, vol. 9, pp. 1-17, 1995.

[18] S.J. Cox, “A Speaker Adaptation Technique Using Linear Regression”, *Proc. of ICASSP*, pp. 700-703, May 1995

[19] C.J. Legetter and P.C. Woodland, “Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models”, *Computer Speech and Language*, vol.9, pp. 1-17, 1995.

[20] M.J.F. Gales and P.C. Woodland, “ Mean and Variance Adaptation within the MLLR Framework”, *Computer Speech and Language*, vol. 10, pp. 249-264.

[21] M.J.F. Gales, “Maximum Likelihood Linear Transformations for HMM-based Speech Recognition”, *Computer Speech and Language*, vol. 12, pp. 75-98.

[22] R.Kuhn, P.Nguyen, J.C. Junqua, N.Niedzielski, S.Fincke and M.Contolini, “Eigenvoices for Speaker Adaptation”, *Proc. of ICSLP* pp. 1771-1774, 1998.

[23] R.Kuhn, P.Nguyen, J.C. Junqua, and L.Goldwasser, “Eigenfaces and Eigenvoices: Dimensionality Reduction for Specialized Pattern Recognition”, *IEEE Signal Proc. letter* 2000.

[24] R.Kuhn, Jean-Claude Junqua, Patrick Nguyen, and Nancy Niedzielski, “Rapid Speaker Adaptation in Eigenvoice Space”, *IEEE Trans. Speech and Audio Proc.*, vol. 8, pp. 695-707, Nov. 2000.

[25] Roland Kuhn, P.Nguyen, J.-Claude Junqua, R.Boman, N.Niedzielski, S.Fincke, K.Field and M.Contolini, “Fast Speaker Adaptation using A-Priori Knowledge”, *Proc. of ICASSP*, pp. 749-752, May 1999.

[26] Yu Tsao, Shang-Ming Lee, and Lin-Shan Lee, “Segmental Eigenvoice With Delicate Eigenspace for Improved Speaker Adaptation”, *IEEE Trans. Speech and Audio Proc.*, vol. 13, pp. 399-411, May. 2005.

[27] Brian Mak, James T. Kwok, and Simon Ho, “Kernel Eigenvoice Speaker Adaptation”, *IEEE Trans. Speech and Audio Proc.*, vol. 13, pp. 984-992, Sept. 2005.

[28] Brian Kan-Wing Mak, Roger Wend-Huu Hsiao, Simon Ka-Lung Ho, and James T. Kwok, “Embedded Kernel Eigenvoice Speaker Adaptation and Its Implication to Reference Speaker Weighting”, *IEEE Trans. Audio, Speech and Lang. Proc.*, vol. 14, pp. 1267-1280, July. 2006.

[29] Lawrence Rabiner and Biing-Hwang Juang, “Fundementals of Speech Recognition”, Pentice Hall PTR 1993.